# Pruebas del funcionamiento del Árbol Binario de Búsqueda

Código:

class Nodo:

    def \_\_init\_\_(self, valor):

        self.valor = valor

        self.derecha = None

        self.izquierda = None

class Arbol:

    def \_\_init\_\_(self):

        self.raiz = None

    def insertar(self, valor):

        self.raiz = self.\_insertar\_recursivo(self.raiz, valor)

    def \_insertar\_recursivo(self, nodo, valor):

        if nodo is None:

            return Nodo(valor)

        if valor < nodo.valor:

            nodo.izquierda = self.\_insertar\_recursivo(nodo.izquierda, valor)

        else:

            nodo.derecha = self.\_insertar\_recursivo(nodo.derecha, valor)

        return nodo

    def buscar(self, valor):

        return self.\_buscar\_recursivo(self.raiz, valor)

    def \_buscar\_recursivo(self, nodo, valor):

        if nodo is None:

            return False

        if nodo.valor == valor:

            return True

        if valor < nodo.valor:

            return self.\_buscar\_recursivo(nodo.izquierda, valor)

        else:

            return self.\_buscar\_recursivo(nodo.derecha, valor)

    def imprimir\_inorder(self, nodo):

        if nodo is not None:

            self.imprimir\_inorder(nodo.izquierda)

            print(nodo.valor)

            self.imprimir\_inorder(nodo.derecha)

## Prueba 1

Texto

Descripción generada automáticamente

La primera línea de código crea una nueva instancia de la clase Arbol, que representa un árbol binario de búsqueda vacío.

Se insertan los siguientes valores en el árbol: 5, 2, 1, 8, 4, 6.

El árbol se organiza en un orden en el que el nodo izquierdo de un nodo tiene un valor menor que el nodo, y el nodo derecho tiene un valor mayor.

Se realiza un recorrido en orden, que consiste en visitar el subárbol izquierdo, el nodo y luego el subárbol derecho.

El resultado de imprimir el árbol en orden debería ser: 1, 2, 4, 5, 6, 8.

Se realiza una búsqueda para el valor 1. El valor 1 está presente en el árbol, por lo que el resultado será True (se encuentra en el árbol).

Luego, se realiza una búsqueda para el valor 9, que no está presente en el árbol. El resultado será False.

## Prueba 2

Texto

Descripción generada automáticamente

La primera línea de código crea una nueva instancia de la clase Arbol, que representa un árbol binario de búsqueda vacío.  
  
Se insertan los siguientes valores en el árbol: 10, 3, 15, 12, 18, 6.  
  
El árbol se organiza en un orden en el que el nodo izquierdo de un nodo tiene un valor menor que el nodo, y el nodo derecho tiene un valor mayor.  
  
Se realiza un recorrido en orden, que consiste en visitar el subárbol izquierdo, el nodo y luego el subárbol derecho.  
  
El resultado de imprimir el árbol en orden debería ser: 3, 6, 10, 12, 15, 18.  
  
Se realiza una búsqueda para el valor 15. El valor 15 está presente en el árbol, por lo que el resultado será True.  
  
Luego, se realiza una búsqueda para el valor 7, que no está presente en el árbol. El resultado será False.